ライントレースロボット

OAKS－LABO

デモプログラム(0klabo_3 & 0klabo_4)の使い方

Ｖｅｒ2.0

ＴＭ統合環境　＋　Ｎｃ30ｗａ　Ｖｅｒ４.００　Ｒｅｌｅａｓｅ２対応版

オークス電子株式会社

目　　次

1.デモプログラムの構成・・・・・・・・・・・・・・・・・・・・・・・・・・・・2

2.デモプログラムの走行・・・・・・・・・・・・・・・・・・・・・・・・・・・3
　（1）モード選択スイッチの割り当てと機能 ・・・・・・・・・・・・・・・・3
　（2）走行テスト手順 ・・・・・・・・・・・・・・・・・・・・・・・・・・4

3.デモプログラムの概要・・・・・・・・・・・・・・・・・・・・・・・・・・・7
　（1）ソフトウェアタイマ方式　・・・・・ ・・・・・・・・・・・・・・・7

4.ＴＭ統合化開発環境での使い方・・・・・・・・・・・・・・・・・・・・・・・9
　（1）インストールと注意点　・・・・・ ・・・・・・・・・・・・・・・・9
　（2）エントリー版（新バージョン）の仕様について・・・ ・・・・・・１０
　（3）ＴＭ統合化開発環境の使い方 ・・・・・・・・・・・・・・・・・・１１

5.割り込み処理・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・１３

6.割り込みベクタテーブルへの登録　・・・・・・・・・・・・・・・・・・・１４

7．便利な関数の活用 ・・・・・・・・・・・・・・・・・・・・・・・・・・１６

1.デモプログラムの構成

ディレクトリ「Ｄｅｍｏ２」に入っている「Ｏｋｌａｂｏ＿３」ファイルは、デバッガ「ＫＤ３０」で使用するためのデバッグ用ファイルです。また、ディレクトリ「Ｄｅｍｏ２＿ＲＯＭ」に入っている「Ｏｋｌａｂｏ＿４」ファイルは、フラッシュＲＯＭに書き込むためのＲＯＭ化用ファイル「Ｏｋｌａｂｏ＿４．ｍｏｔ」を作成します。商品には表１で示されるファイルが入っています。

表１　「Ｄｅｍｏ２」ディレクトリのファイル一覧

ディレクトリ名	ファイル名	内　　容
Deno2	Oklabo_3.c	デバッグ用タイマ割込み方式ソースプログラム。
	Oklabo_3.a30	コンパイル出力アセンブラファイル。
	Oklabo_3.r30	アセンブラ出力ファイル。
	Oklabo_3.map	マップファイル。
	Oklabo_3.lst	リストファイル。
	Oklabo_3.x30	デバッグ用出力ファイル。
	Oklabo_3.tmk	プロジェクトファイル。
	Oklabo_3.mak	コンパイル用メイクファイル。
	M16io.h	I/O 記述ヘッダファイル。
	Lcdtime.h	LCD 表示とタイマ用ヘッダファイル。
	ncrt0.a30	スタートアップルーチンファイル。
	sect30.inc	スタートアップルーチン用インクルードファイル
	lcc.bat	コンパイル用バッチファイル。
Demo2_rom	Oklabo_4.c	ROM 化用タイマ割込み方式ソースプログラム。
	Oklabo_4.a30	コンパイル出力アセンブラファイル。
	Oklabo_4.r30	アセンブラ出力ファイル。
	Oklabo_4.map	マップファイル。
	Oklabo_4.lst	リストファイル。
	Oklabo_4.x30	**********
	Oklabo_4.mot	ROM 化用オブジェクトファイル。
	Oklabo_4.tmk	プロジェクトファイル。
	Oklabo_4.mak	コンパイル用メイクファイル。
	M16io.h	I/O 記述ヘッダファイル。
	Lcdtime.h	LCD 表示とタイマ用ヘッダファイル。
	ncrt0.a30	スタートアップルーチンファイル。
	sect30.inc	スタートアップルーチン用インクルードファイル
	lcc.bat	コンパイル用バッチファイル。

2.デモプログラムの走行

デモプログラム「Ｏｋｌａｂｏ＿３」または「Ｏｋｌａｂｏ＿４」を使用してＯＡＫＳ－ＬＡＢＯを走行させる場合には、コースが必要です。床または卓上が白色の場合には、黒のビニールテープを付録５のコース例を参考に貼って下さい。また、床または卓上が黒色の場合には、白のビニールテープをユーザーズマニュアル付録８のコース例を参考に貼って下さい。

コースはなるべく段差を少なくして下さい。段差があると車輪の空回りやボールキャスタの引っ掛かりなど、ロボットがうまく動かない場合があります。また、紙でコースを作ると車輪のゴムに紙の繊維が付着してグリップ力が無くなり空回りの原因になります。

走行テストの前にモータ電流とラインセンサの調整をしておきます。また、付属のバッテリパックの充電は、できる限り専用の充電器を使用して下さい。もし、手元に無い場合には、ＤＣ１４Ｖ～１６Ｖ、２００～３００mA程度（定格１１０mA　１５時間）でゆっくり充電して下さい。

(1)、モード選択スイッチの割り当てと機能

モード選択スイッチは、表２のように４つのスイッチの状態によって、速度の設定、オールセンスの判定による停止または通過の設定､白ライン走行または黒ライン走行の設定ができます。

① スイッチ１番と２番の機能

スイッチ１番と２番の４通りの組み合わせによって、表４のように低速、中低速、中高速、高速の４段階に設定できます。設定変更は、電源を投入する前またはＬＣＤ表示器に表示されるスタートまでの１０秒間以内に行います。

② スイッチ３番の機能

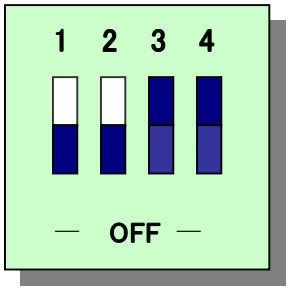
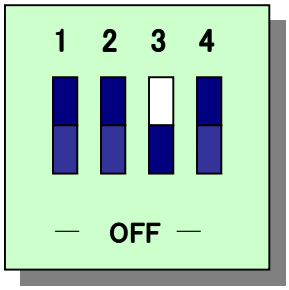
３番のスイッチは、オールセンス時での動作を選択します。“ＯＮ”で使用する場合には、オールセンスの無いコース(交差点の無いコース)で走行して下さい。ストップライン(４つのセンサが同じに消灯(黒ライン走行)または点灯(白ライン走行))で停止します。

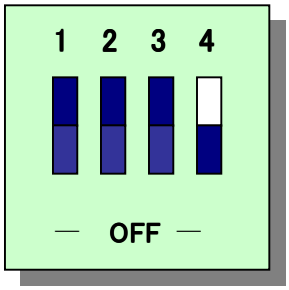
スイッチを“ＯＦＦ”で使用する場合には、交差点などオールセンスのあるコースを走行させることができます。設定変更は、電源を投入する前またはＬＣＤ表示器に表示されるスタートまでの１０秒以内に行います。

① スイッチ４番の機能

４番スイッチは､“ＯＮ”にすると白地に黒ラインのコースで走行できます｡また、ＯＦＦ”で使用すると黒地に白ラインで走行できます。設定変更は、電源を投入する前またはＬＣＤ表示器に表示されるスタートまでの１０秒以内に行います。

表２　モード選択スイッチの割り当てと機能

<table>
<tr><th>モード選択スイッチ</th><th>機　　能</th></tr>
<tr><td> 
</td><td>モード選択スイッチの１番と２番は走行速度を選択します。
<table>
<tr><th>1</th><th>2</th><th>LCD 表示</th><th>走行速度</th></tr>
<tr><td>OFF</td><td>OFF</td><td>Lo</td><td>低速</td></tr>
<tr><td>OFF</td><td>ON</td><td>Ml</td><td>中低速</td></tr>
<tr><td>ON</td><td>OFF</td><td>Mh</td><td>中高速</td></tr>
<tr><td>ON</td><td>ON</td><td>Hi</td><td>高速</td></tr>
</table></td></tr>
<tr><td> 
</td><td>モード選択スイッチの３番は通過と停止を選択します。

ＯＮ・・・減速・停止します。(LCD 表示:　“St”)
オールセンスの無いコースで、ゴール線で停止。
ＯＦＦ・・通過します。(LCD 表示:　“Pa”)
交差点等、オールセンスが在るコース。</td></tr>
<tr><td> 
</td><td>モード選択スイッチの４番は白ラインと黒ラインを選択します。

ＯＮ・・・白地に黒ラインの場合。
(LCD 表示:　“Bk”)
ＯＦＦ・・黒地に白ラインの場合。
(LCD 表示:　“Wh”)</td></tr>
</table>

(2)、走行テスト手順

① 電源スイッチを“ＯＮ”またはリセットスイッチを押すと図１に示される表示が数秒間表示されます。

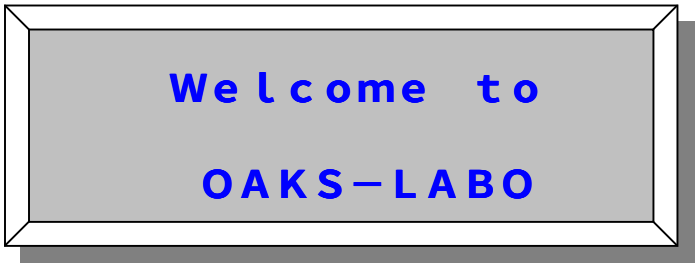


図１ メッセージの表示

② 数秒後に図２で示されるモータ電流調整の為の表示に切り替わります。励磁遮断スイッチが共に“ＯＦＦ”にすると「Ｌ＝０００mA　Ｒ＝０００mA」が表示されます。左側の励磁遮断スイッチを“ＯＮ”にしてから左モータ用の励磁電流調整ボリュームを廻して、図２のように約Ｌ＝５００mAとなるように調整します。電流値は±４０mA程度常に変動しています。故障ではありません。次に右側の励磁遮断スイッチを“ＯＮ”にしてから右モータ用の励磁電流調整ボリュームを廻して、図２のように約Ｒ＝５００mAとなるように調整します。電流値は同様に±４０mA程度常に変動しています。故障ではありません。調整が終了後、スタートスイッチを押して下さい。この場合、画面が切り替わる迄しばらく押し続けて下さい。

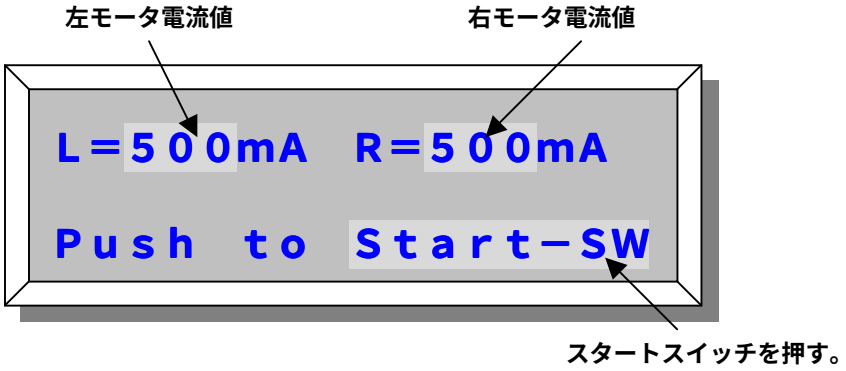


図２　モータ電流の調整表示

②　スタートスイッチを押すと、図３で示されるモード表示画面に切り替わります。上段には、走行開始までの残り時間が表示されます。時間は、１０秒から開始し、０秒になると走行を開始します。下段は、表４で表記されるように、モード選択スイッチの状態を表示します。変更を行う場合には、走行カウンタが０秒になる迄にモード選択スイッチで行って下さい。

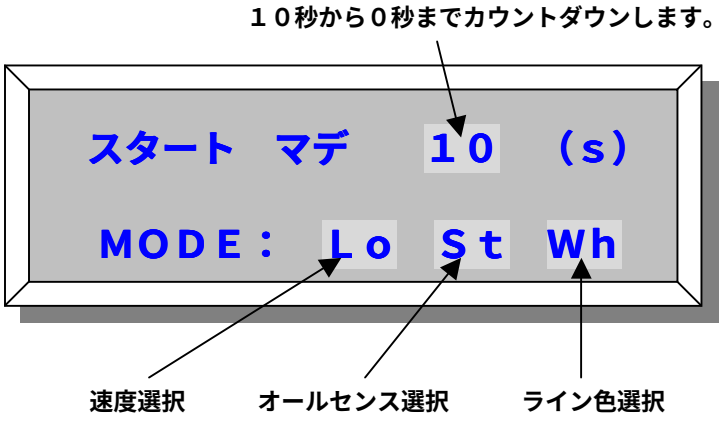


図３　モード表示画面

③　　走行カウンタが０秒になると、 図４で示される走行表示画面に切り替わります。

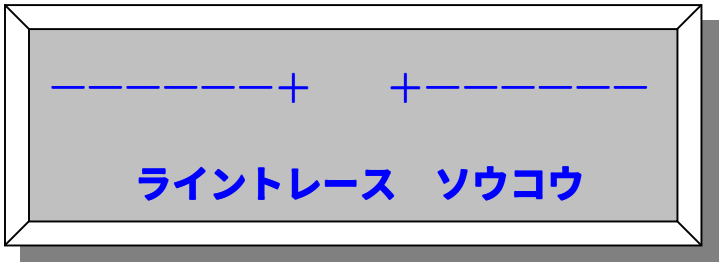


図４　走行表示画面

④　　モード選択スイッチで走行開始前に、予め停止ラインの減速・停止の設定がされていると、図５で示されるように、走行開始から停止迄の所要時間が表示されます。

再び走行させる場合には、スタートスイッチを押して下さい。③項の画面表示に戻ります。

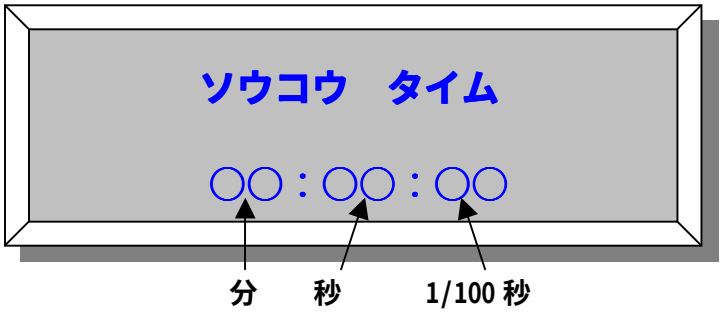


図５　走行表示画面

⑤　　左接触センサが障害物に接触すると、図６で示されるように、左側が障害物に接触した表示を行い、急停止します。その後、数十センチ程度バック走行を行い完全停止します。

再び走行させる場合には、コースに戻し、スタートスイッチを押して下さい。③項の画面表示に戻り再走行を開始します。

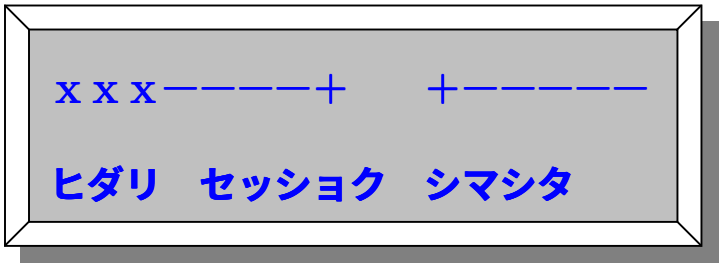


図６　左側接触の表示画面

⑥　　右接触センサが障害物に接触すると、図７で示されるように、右側が障害物に接触した事を知らせる表示を行い、急停止します。その後、数十センチ程度バック走行を行い完全停止します。

再び走行させる場合には、コースに戻し、スタートスイッチを押して下さい。③項の画面表示に戻り再走行を開始します。

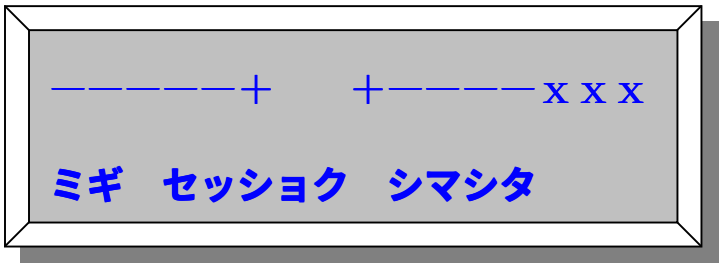


図７　右側接触の表示画面

3.デモプログラムの概要

(1) タイマ割り込み方式

デモプログラム「Ｏｋｌａｂｏ＿３」と「Ｏｋｌａｂｏ＿４」は、左右のモータをそれぞれ別々のタイマ割り込みによって回転速度を制御する方式です。図８はタイマ割り込み方式のフローチャートです。

この方式は、プログラムにタイマ割り込みの知識が必要となりますが、左右のモータに割り当てたタイマの時間定数を変更することで独立して回転速度を制御できます。割込みのあり、なしの判断に割込みフラグを使用します。タイマＡをタイマモードに設定し、その割込み先をモータ駆動にします。時間定数値をセットし、割込みがある迄待ちます。割込みが発生し、モータを駆動してフラグを１にセットします。時間定数に加速データを与える事により、より高速走行が可能となります。

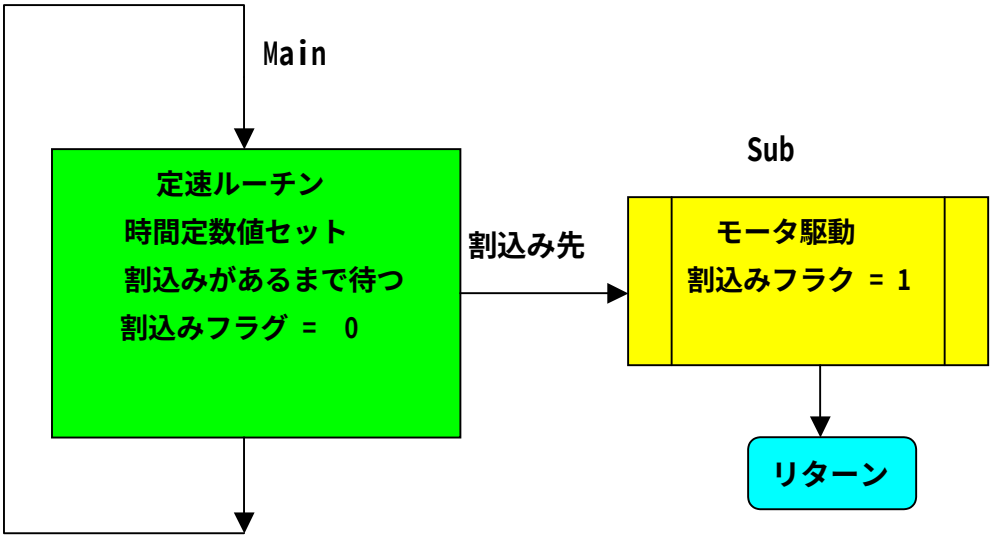


図８ タイマ割込み方式でのモータ駆動

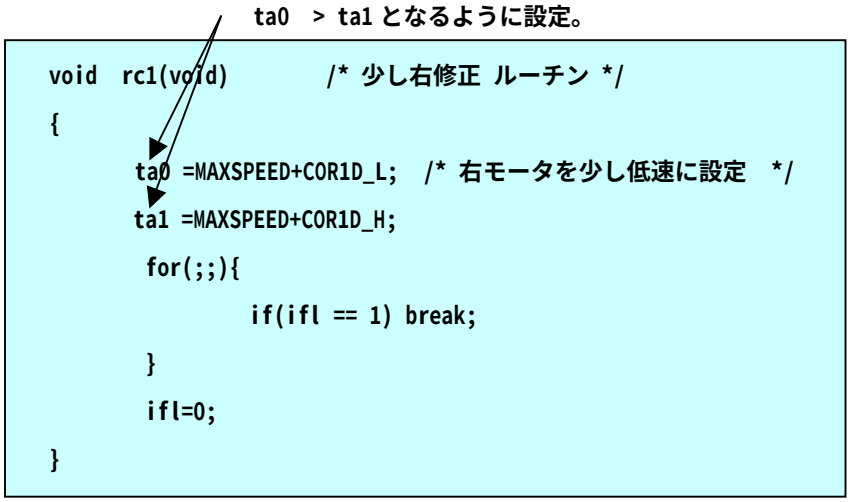
図９（ａ）は、図８図で示されるフローチャートをＣ言語で記述したプログラム例です。右モータの時間設定を行うタイマ変数がｔａ０です。また、左モータの時間設定を行うタイマ変数はｔａ１です。直進する場合には、2 つのタイマ変数に規定値 MAXSPEED を設定します。右モータ駆動割込みが終了すると割込みフラグ「ｉｌｒ」を”１”にします。また、左モータ駆動割込みが終了すると割込みフラグ「ｉｆｌ」を”１”にします。したがって、無限ループで何れかのフラグが”１”になるまで待ちます。最後に 2 つのフラグを”０”に戻して割込みを可能にします。

規定値

```
Void  noncr(void)  /* 直進 ルーチン */
{
        ta0 = MAXSPEED;    /* 右定速度回転  */
        ta1 = MAXSPEED;    /* 左定速度回転 */
         for(;;){
                 f(ifr == 1 | ifl == 1) break;
         }
         ifr=0;
         ifl=0;
}
```

（ａ）直進ルーチン

図９（ｂ）は、ロボットを少し右に修正する場合のプログラム例です。直進する場合には、右モータのタイマ変数には規定値 MAXSPEED と可変値 COR１D__L を加えます。また、左モータのタイマ変数は規定値 MAXSPEED と可変値 COR１D__H を加えます。
可変値は、COR１D__L　＞　COR１D__H となる値を与えるので、右モータのタイマ値が左モータのタイマ値よりも大きくなります。したがって、右モータの回転速度が左モータの回転速度よりも遅くなり、ロボットは右方向に進みます。
左右２つの可変値を組み合わせると、方向を自由に変える事が可能になります。



（ｂ）少し右修正ルーチン

図９　タイマ割込み方式のプログラム記述例

４．ＴＭ統合化開発環境での使い方

ＯＡＫＳ－ＬＡＢＯ用ＲＳ-２３２Ｃ無線ユニットには、最新のＴＭ統合開発環境「ＴＭＶ３．１１」が付属しています。このＴＭ統合化開発環境を用いることにより、バッチファイルまたはメイクファイルによりＤＯＳ窓上でコンパイルを行っていた方式からＷｉｎｄｏｗｓ上でエディタの起動、コンパイラ、デバッガの起動等、統合操作を可能にしました。

（１）インストールと注意点

付属のＣＤ－Ｒには、ＴＭ統合化開発環境「ＴＭＶ３．１１」とコンパイラ「ＮＣ３０ＷＡ　Ｖ４．００　Ｒｅｌｅａｓｅ ２(エントリー版)」が入っています。既に、ライントレースロボット「ＯＡＫＳ－ＬＡＢＯ」に添付されているコンパイラ「ＮＣ３０ＷＡ　Ｖ３．００」がインストールしている場合、インストール中の「コンポーネントの選択」画面で「インストール先ディレ

クトリ」項目を「参照（R)」を使用して別のインストール先を指定してインストールをして下さい。もし、同じディレクトリにインストールする場合、必ず、既にインストールされているＮＣ３０WAをアンインストールした後で、本製品のＮＣ３０WAをインストールして下さい。

ＯＡＫＳ-ＬＡＢＯに添付されている全てのデモサンプルプログラムを旧バージョン「ＫＮＣ３０　Ｖ３.００」でコンバイルするとエラーが発生します。したがって、本製品に入っている全てのデモサンプルプログラムは、旧バージョンではコンパイルできません。

（２）エントリー版（新バージョン）の仕様について

① プログラムの記述について

旧バージョンのプログラムをエントリー版でコンバイルする際、表３示される語の前にアンダースコア（__）を付加して下さい。

表３　新旧バージョンの記述相違一覧

KNC30 旧バージョン版	NC30 エントリー版
ｉｎｌｉｎｅ	__ｉｎｌｉｎｅ
ｎｅａｒ	__ｎｅａｒ
ｆａｒ	__ｆａｒ
ａｓｍ ()	__ａｓｍ ()

② ＴＭ統合開発環境「ＴＭＶ３.１１」について

- インスペクタ機能はご使用になれません。
- ブラリプロジェクトは作成できません。

② ソフトウェアおよびユーティリティについて

下記に示すソフトウェアおよびユーティリティはご使用になれません。

表４　使用できないソフトウェアおよびユーティリティー覧

ソフトウェア	STK ビューワ，MPA ビューワ，アセンブルオプティマイザ (aopt30)，ライブラリアン(lb30)，構造化記述アセンブラ (pre30),] 標準関数ライブラリソースファイル
ユーティリティ	utl30(SBDATA 宣言&SPECIAL ページ宣言ユーティリティ

④　ＮＣ３０とＡＳ３０のオプションについて

マニュアルに記載されている下記のオプションはご使用になれません。

表５　使用できないＣコンパイラオプション一覧

Ｃコンパイラ（ＮＣ３０）	
デバッグ用オプション	genter　-gno_reg
最適化オプション	-O[1-5], -OR, -OS, -Oconst(-OC), -Ono_bit(-ONB),-Ono_break_source_debug(-ONBSD), -Ono_float_const_fold(-ONFCF), -Osp_adjust(-OSA), -Ostack_frame_align(-OSFA), Oloop_unroll(-OLU), Ono_asmopt(-ONA), -Ono_logical_or_combine(-ONLOG), -Ocompare_byte_to_word(-OCBTW), -Ono_stdlib
生成コード変更オプション	-finfo, -fuse_DIV(-fUD), -fansi, -fnear_ROM(-fNROM), -fsmall_array(-fSA), -fno_align(-fNA)
アセンブル、リンクオプション	-as30, -ln30
その他のオプション	-dsource(-dS), -dsource_in_list(-dSL)

表６　使用できないアセンブラオプション一覧

アセンブラ（ＡＳ３０）	
オプション	-finfo, -P, -M

（３）ＴＭ統合化開発環境の使い方

ＴＭ統合化開発環境「ＴＭ」を使用する場合、ライントレースロボット「ＯＡＫＳ－ＬＡＢＯ」に添付されているコンパイラ「ＫＮＣ３０」(旧バージョン)ではご使用になれません。「ＴＭ」を使用する場合には、本製品に入っているエントリー版コンパイラ「「ＮＣ３０」(新バージョン)を使用します。「ＴＭ」を起動すると細長いプロジェクトバーが表示されます。マウスによって右端を縮めると図１０で示されるフローティング状態のプロジェクトバーが表示されます。

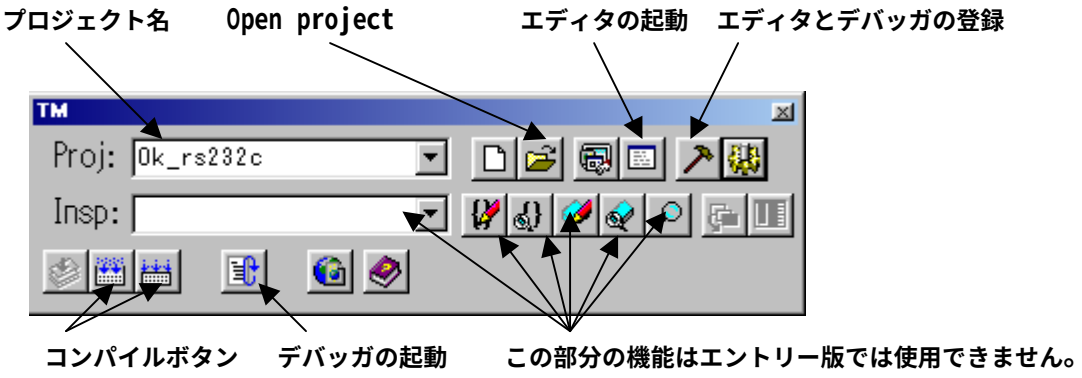


図１０　プロジェクトバー

① エディタとデバッガの登録

最初に (ツールの登録)ボタンをクリックし、使い慣れたエディタとリモートデバッガ「ＫＤ３０」を登録します。

ここで登録しないと、図１９図の (エディタの起動)ボタンと (デバッガの起動)ボタンが使用できません。

② プロジェクトファイルのロード

ＴＭ統合化開発環境「ＴＭ」を用いてＷｉｎｄｏｗｓ上でコンパイルを行うには、プロジェクトファイルを作成しなければなりません。新規のユーザプログラムからプロジェクトファイルを作成しコンパイルする方法は、付属のユーザズマニュアル（ＰＤＦファイル）をご覧下さい。本製品のデモサンプルプログラムにはプロジェクトファイル「Ｏｋ＿ｒｓ232ｃ．ｔｍｋ」が入っています。

プロジェクトファイルをロードするには図１９で示される (プロジェクトを開く)ボタンをクリックします。開くウインドウから付属のＣＤ－Ｒからコピーしたドライブとディレクトリを選び「Ｏｋ＿ｒｓ232ｃ．ｔｍｋ」をロードします。図１９で示されるプロジェクトウインドウにファイル名が表示されます。

③ コンパイル方法

プロジェクトで設定したソースプログラム「Ｏｋ＿ｒｓ232ｃ．ｃ」をコンパイルするには (ビルドまたはリビルド)ボタンをクリックします。ウインドウにコンパイルメッセージが表示されます。スクロールバーによって前後のメッセージを読むことが可能です。

④ エディタの起動

(ツールの登録)ボタンによって前もって使い慣れたエディタを登録しておくと、(エディタの起動)ボタン でエディタが起動します。ソースプログラムをロードしてプログラムを修正できます。

⑤ デバッガの起動

(ツールの登録)ボタンによってリモートデバッガを登録しておくと、 (デバッガの起動)ボタンをクリックするとリモートデバッガ「ＫＤ３０」が起動できます。

図１０において、２段目のインスペクタとその関連機能は、エントリー版では使用できません。また、ＴＭ統合化開発環境「ＴＭ」の詳しい使用方法は、本製品のＣＤ－Ｒに入っているユーザーズマニュアルをご覧下さい。

５.割り込み処理

Ｍ１６Ｃ／６２シリーズのＣＰＵで割り込みをＣ言語関数として記述する手順は次の２つです。

① 割り込み処理関数の定義。

② 割り込みベクタテーブルの登録。

割り込み処理関数の定義は、割り込み関数名に適当な名前を付けて、次のように記述します。

```
＃ｐｒｏｇｒａｍ　ＩＮＴＥＲＲＵＰＴ　割り込み関数名
```

上記のように記述すると、指定した関数の入り口と出口において、通常の関数の手続き以外に、全レジスタの退避、復帰と“ｒｅｉｔ”命令を生成します。割り込み処理関数の型は、引数／戻り値共にｖｏｉｄ型のみ有効です。図１１は、デモサンプルプログラム「Ｏａｋｓｌａｂｏ＿４」で割り込みに使用する為の割り込み処理関数の記述です。

割り込み関数名

```
/************************************************************************
*   割込み関数定義
************************************************************************/
void        int0int();          /* ＩＮＴ０割込み関数       */
    #pragma INTERRUPT  int0int
void        int1int();          /* ＩＮＴ１割込み関数       */
    #pragma INTERRUPT  int1int
void  intr_count(void);         /* ＬＣＤタイマ値割込み関数 */
    #pragma INTERRUPT  intr_count
void  ta3int(void);             /* 秒カウント割込み関数     */
    #pragma INTERRUPT  ta3int
void  rmout(void);              /* 右モータ割込み関数       */
    #pragma INTERRUPT  rmout
void  lmout(void);              /* 左モータ割込み関数       */
    #pragma INTERRUPT  lmout
```

図１１ 「Ｏａｋｓｌａｂｏ＿４」の割り込みの記述

６.割り込みベクタテーブルへの登録

割り込みを正常に使用する為には、割り込み処理関数を定義すると共に割り込みベクタテーブルに登録する必要があります。

割り込みベクタテーブルの変更は、次の手順で行います。

① 割り込み処理関数名を疑似命令“.ｇｌｂ”で外部定義します。

② 使用する割り込みのダミー関数“ｄｕｍｍｙ＿ｉｎｔ”を、割り込み処理関数名に変更します。

図１２は、スタートアップルーチン「ｎｃｒｔ０．Ａ３０」に記述されている見出し文「Interrupt section start」内に割込み処理関数を記述したものです。外部定義では次のように必ずアンダーバーを付けて記述して下さい。

```
.glb        _rmout
```

また、タイマＡ０を割り込みで使用する場合には、図１３に示されるインクルードファイル「ｓｅｃｔ３０．ｉｎｃ」のベクターテーブル見出し名「variable vector section」２１番のダミー関数“ｄｕｍｍｙ＿ｉｎｔ”を次のように必ずアンダーバーを付けて記述して下さい。

```
.lword      _rmout
```

割込み処理関数名の記述

```
;=======================================================
; Interrupt section start
;-------------------------------------------------------
          .insf     start,S,0
          .glb      _rmout
          .glb      _lmout
          .glb      _intr_count
          .glb      _ta3int
          .glb      _int0int
          .glb      _int1int
          .glb      start
          .section  interrupt
start:
```

図１２　スタートアップルーチンに割込み処理関数名を記述する例

```
else
      .lword    dummy_int          ; vector 0 (BRK)
      .org      (VECTOR_ADR +44)
      .lword    dummy_int          ; DMA0 (for user)
      .lword    dummy_int          ; DMA1 2 (for user)
      .lword    dummy_int          ; input key (for user)
      .lword    dummy_int          ; AD Convert (for user)
      .org      (VECTOR_ADR +68)
      .lword    dummy_int          ; uart0 trance
      .lword    dummy_int          ; uart0 receive
      .byte     6bh                ; uart1 trance
      .byte     0cbh
      .byte     0fh
      .byte     00h
      .byte     6bh                ; uart1 receive
      .byte     0cbh
      .byte     0fh
      .byte     00h
      .lword    _rmout             ; TIMER A0 タイマ A0
      .lword    _lmout             ; TIMER A1 タイマ A1
      .lword    _intr_count        ; TIMER A2 タイマ A2
      .lword    _ta3int            ; TIMER A3 タイマ A3
      .lword   dummy_int           ; TIMER A4 (vector 25)
      .lword    dummy_int          ; TIMER B0 (vector 26)
      .lword    dummy_int          ; TIMER B1 (vector 27)
      .lword    dummy_int          ; TIMER B2 (vector 28)
      .lword    _int0int           ; INT0     (vector 29)
      .lword    _int1int           ; INT1     (vector 30)
      .lword    dummy_int          ; INT2     (vector 31)
.endif
```

割込み処理関数名の記述

図１３　ベクターテーブルの記述例

７.便利な関数の活用

ＬＣＤ表示関数

デモプログラムが使用しているヘッタファイル「ｌｃｄｔｉｍｅ＿１．ｈ」にはＬＣＤを表示させるのに便利な関数が記述されています。

① 表示位置関数「ｇｏｔｏｘｙ(ｘ，ｙ)」
表示位置関数の括弧内に記述するＸは、表示する桁位置で０から１５まで表示位置を記述します。また、Ｙは、表示する行位置で上段は０、下段は１を記述します。

ｇｏｔｏｘｙ（桁位置，行位置）

② 表示関数「ｌｃｄｐｒｉｎｔ（配列名)」
表示関数の括弧内には直接に表示データを記述することはできません。一旦配列で記述してから括弧内に配列名で記述します　。

ｌｃｄｐｒｉｎｔ（配列名）

図１３は、ＯＡＫＳ－ＬＡＢＯの電源を投入後、最初に表示するメッセージの表示プログラムです。表示データは「titl_1」と「titl_2」 に配列で記述します。配列「Titl_1」のデータは１行目０桁位置から表示しのか。また、配列「titl_2」のデータは２行目の０桁位置から表示します。

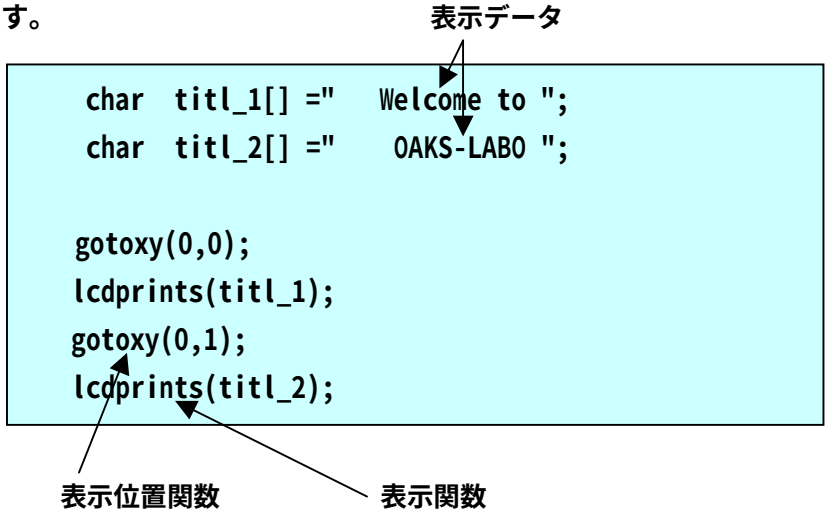


図１３　ＬＣＤのメッセージ表示例

●ご注意

(1)　本書の内容の一部または全部を無断転載することは禁止されています。

(2)　本書の内容は将来予告なしに変更することがあります。

(3)　本書は、内容について万全を期して作成致しましたが、万一ご不振な点や誤り、記載漏れなどお気づきなことがありましたら、お買い上げの販売代理店にご連絡ください。

(4)　運用した結果の影響につきましては、(3)項にかかわらず責任を負いかねますのでご了承ください。

●参考文献

(1)「マイコン制御ロボットの製作」横山直隆著　シータスク発行

(2)「Ｙ－ＲＯＢＯ取扱説明書」　サン・マイテック社

(3)「コンパイラ　ＮＣ30ＷＡ Ver4.0」 三菱電機

●商標

Ｗｉｎｄｏｗｓ９５／９８／ＭＥ／ＮＴは、米国マイクロソフト社の登録商標です。

デモプログラム(Oklabo_３ & Oklabo_４)の使い方

２００１年１０月２３日　第二冊発行

編集・発行　　***オークス電子株式会社***

〒101-0025 東京都千代田区神田佐久間町３－２１－３

第一千代田ビル３Ｆ

TEL (03)3863-1121　FAX (03)3863-1130

http://WWW.oaks-ele.com